

# Comandos Linux/Bash

*dir (directorio/s)*

*cmd (comando/s)*

*^(CTRL)*

*: (ESC)*

*' ' (SPACE)*

1. **clear** (limpiar pantalla)
2. **date** (fecha)
3. **cal** (calendario): **cal mes año** /Ej. **cal 12 2021**
4. **bc** (calculadora), salir (**^d**)
5. **man** (manual), salir (**q**) /Ej. **man cal**
6. **info** (similar a manual)
7. **-help** (ayuda) /Ej. **cal -help**
8. **whatis** (ayuda resumida) /Ej. **whatis cal**
9. **pwd** (ver directorio de trabajo)
10. **last -2** (ver las 2 últimas conexiones)
11. **whereis archivo** (ubicar un archivo)
12. **apropos archivo** (visualizar donde se “habla” de un comando o texto en el sistema)
13. **stat archivo** (visualizar el estado de un archivo)
14. **l** (lista de los directorios)
15. **ls** (lista de los directorios)
  - ls -a** (listado de los directorios ocultos y los directorios de referencia)  
*[los directorios ocultos empiezan por .]*
  - ls -A** (listado de los directorios ocultos sin los directorios de referencia)  
*[los directorios ocultos empiezan por .]*
  - ls -l** (listado largo)
  - ls -la** (listado largo de los directorios ocultos)
  - ls -R** (recursiva, entra en todos los subdirectorios)
  - ls -r** (al revés que ls)
  - ls a\*** (buscar archivos que empiecen con la **a**)
  - ls \*a\*** (buscar archivos que contengan la **a**)
  - ls \*a** (buscar archivos que terminen con la **a**)
  - ls -d a\*** (buscar archivos que empiecen por la **a** sin entran en los **dir**)
  - ls \*a\*b\***
  - ls \*ab\*o**
  - ls [ab]\*** (buscar archivos que empiecen con **a** y **b**)
  - ls [ab]c\*** (buscar archivos que empiecen por **a** y **b** y que sigan con la **c**)
  - ls [a-z]\*** (buscar archivos que empiecen por alguna letra de la **a** a la **z**)
  - ls [0-9]\*** (buscar archivos que empiecen por algún número del **0** al **9**)
  - ls [!2]\*** (buscar archivos que **no** empiecen por un **2**)
  - ls ?2\*** (buscar archivos que empiecen por un **2** en **segundo** lugar)
  - ls -s** (size, tamaño archivos en bloques)
  - ls -1** (ver todo en una columna)
  - ls -F** (clasificación de los tipos de ficheros)
  - ls -i** (inodos, número exclusivo de cada fichero)

- ls -lis** (listar viendo toda la información del fichero)
- 16. **~** (mi home)
- 17. **cd** (cambiar de directorio) /Ej. **cd /** (raíz)  
**cd** [solo] o **cd ~** (salta al home)  
**cd -** (último directorio)  
**cd ..** (ir al directorio padre con ruta relativa)  
**cd ../..** (ir al directorio padre del padre con ruta relativa)
- 18. **mkdir** (crear directorio) **mkdir nombre del dir** /Ej. **mkdir uno**  
**mkdir -p** (crear sucesorios directorios) /Ej. **mkdir -p uno/dos/tres**
- 19. **pwd; date; cal 10 2021** (órdenes sucesorias)
- 20. **>** (dirección de salida, escribe y sobreescribe)  
/Ej. (**pwd; date; cal 10 2021**) **> datos.txt**
- 21. **<** (dirección de entrada)
- 22. **>>** (direccionamiento de salida doble, añade)
- 23. **file** (tipo y formato de un archivo)
- 24. **more** (visualizar un archivo pudiendo desplazarse hacia abajo, actua también como filtro, actua como un cat)
- 25. **less** (visualizar un archivo pudiendo desplazarse hacia abajo y arriba, actua también como filtro, actua como un cat)
- 26. **history** (historial de comandos)
- 27. **history número** (historial de los **n** últimos comandos)
- 28. **|** (barra de filtro o tubería, ficheros de tipo pipe)
- 29. **history | more** (lista de historial de comandos)  
[luego desplazarse con intro o espacio]  
**^C** (abortar, cortar listado de history | more)
- 30. **history | less** (lista de historial de comandos)  
[intro, espacio o flechas]  
**q** (abortar, cortar listado de history | less)
- 31. **!número** (recuperar un comando del history) /Ej. **!15**
- 32. **!!** (recuperar último comando del history)
- 33. **cat > datos.txt** (crear el archivo **datos.txt** de forma interactiva) **^D** (para salir)  
**cat datos.txt** (visualizar el fichero **datos.txt**)  
**cat datos.txt | more** o **more datos.txt**  
**cat -n datos.txt** (numera las líneas del fichero **datos.txt**)
- 34. **tac datos.txt** (ver el fichero al revés **datos.txt**)
- 35. **echo** (escribe en pantalla)  
**echo {a..z}** (escribe de la **a** a la **z**)  
**echo {quiero,tomar,café}** o **echo "quiero tomar café"** (escribe **quiero tomar café**)  
**echo \$?** (pregunta como salio el comando anterior; si responde 0 no hubo ningún error, si responde otro nº, existió un error)  
**echo \$BASH\_VERSION** (visualizar la variable de entorno **BASH\_VERSION**)
- 36. **rm** (eliminar archivos)
- 37. **rmdir** (eliminar directorios)

38. **rm -r** (eliminar directorios que no estén vacíos)
39. **mv origen destino / mv nombre1 nombre2 / mv origen destino nombre2**  
(mover un archivo / renombrar)
40. **file** (información del contenido del archivo)  
**file\*** (información del contenido de todos los archivos)
41. **touch** (crear archivos vacíos) /Ej. **touch [archivo1.txt archivo2.txt]**  
**touch -t YYMMDDhhmmss** (cambiar time)  
**touch -r archivo1 archivo2** (poner la fecha de modif de archivo1 a archivo2)  
**touch -c** (cambiar fecha del fichero si existe poniendo la fecha del sistema)
42. **cp** (copiar)
43. **logname** (cuenta con la que se está conectado)
44. **id** (número de indentificación de todos los usuarios)  
**id -u** (número de indentificación de la cuenta en la que se está conectado)  
**id -nu (nombre de usuario)**  
**id -g** (salida solo el ID de grupo efectivo)  
**id -gn (nombre del grupo principal)**  
**id -a** (imprimir nombre de usuario, UID y todo el grupo al que pertenece el usuario)
45. **whoami** (¿quién soy yo?, cuenta con la que se está trabajando)  
**who am i** (mayor info que whoami)  
**w** (¿quienes están conectados?)  
**who -T** (que terminales tienen activado o desactivado recibir mensajes, lista de quienes están conectados)  
**who -q** (cantidad de usuarios que están conectados)
46. **groups** (cantidad de los grupos)
47. **uname** o **uname -s** (el **sistema operativo** con el que se está a trabajar)  
**uname -m** (microprocesador)  
**uname -r** (release, versión del kernel)  
**uname -v** (versión de la distribución)  
**uname -n** (nodo, nombre terminal)  
**uname -a** (información sobre hardware y el sistema operativo)
48. **sudo** (switch-do, para entrar en directorios en los que se necesiten permisos)
49. **ln (hard link/enlace duro)**  
**ln -s** (enlace/acceso directo **simbólico**)
50. **vi [nombre del fichero]**  
**vim [nombre del fichero] (más funciones)**  
(apuntes)
51. **PERMISOS:**  

	<b>r</b> leer	<b>w</b> escribir	<b>x</b> ejecutar	
	<b>rwx</b>	<b>rw-</b>	<b>r--</b>	
	<b>u</b>	<b>g</b>	<b>o</b>	
<b>0</b>	<b>6</b>	<b>6</b>	<b>4</b>	octal+
<b>especial</b>	<b>110</b>	<b>110</b>	<b>100</b>	binario

## 52. PERMISOS ESPECIALES:

52.1. **t** (sticky bit, protege el contenido de un fichero o directorio que solo podrá ser eliminado por el creador)

T (reemplaza a la x, en los permisos de otros)

52.2. **suid = s** (el que lo ejecute va a tener los mismos permisos que el que creó el archivo)

S (reemplaza a la x, en los permisos de usuario y grupo)

52.3. **sgid = g+s** (todo archivo que tenga activo el SGID, al ser ejecutado, tendrá los privilegios del grupo al que pertenece)

S (reemplaza a las x, en los permisos de grupo)

53. **umask** (máscara del sistema)

**umask máscara** (cambiar máscara del sistema)

54. **chmod archivo n<sup>o</sup>octal** (cambiar configuración del archivo en octal)

**chmod g-w,o+r archivo** (cambio de permisos: quitamos escribir y añadimos leer)

**chmod g=rw archivo** (cambio de permisos: sobreescribimos los permisos añadir leer y escribir)

**chmod -R** (cambio de permisos dentro de un fichero, recursiva)

55. **su** (switch user)

56. **stdir** 0 <

**stdout** 1 > >>

**stderr** 2 2> /Ej. cat listado.txt 2> errores.txt

/Ej. cat listado.txt 2> /dev/null (basura errores)

## 57. COMANDOS EN CORTOCIRCUITO:

57.1. **&&** y

57.2. **||** o

58. **tee** (sobreescribe a un fichero y a display)

**tee -a** (añade a un fichero y a display)

**| tee** (actúa también como filtro)

59. **sort** (ver un fichero ordenado por el primer campo)

**sort -n** (ver un fichero ordenado por n<sup>o</sup>)

**sort -t: -k5 -n empleados.txt** (ordenar numéricamente por el 5<sup>o</sup> campo con delimitador :, de menor a mayor)

**sort -t: -k5 -n -r empleados.txt** (ordenar numéricamente por el 5<sup>o</sup> campo con delimitador :, de mayor a menor)

**sort -t: -k5 -n -r empleados.txt | tee fichero.txt** (ordenar numéricamente por el 5<sup>o</sup> campo con delimitador :, de menor a mayor mandando a )

**| sort** (actúa también como filtro)

60. **wc** (da 3 valores: cant líneas, cant palabras y cantidad de caracteres)  
**wc -l** (cat líneas)  
**wc -w** (cant palabras)  
**wc -c** (cat caracteres)  
**wc -L** (cant caracteres que tiene la línea más larga)  
**| wc** (actúa también como filtro) /Ej. *ls -l | wc -l*
61. **head** (display de 10 primeras líneas de un fichero)  
**head -3 emple\*** (display de 3 primeras líneas de los ficheros que empiecen con emple)  
**| head** (actúa también como filtro) /Ej. *sort -t: -k5 -n empleados.txt | head -4*  
*sort -t: -k5 -n empleados.txt | head -4 | sort -n*
62. **tail** (display de 10 últimas líneas de un fichero)  
**tail -2** (display de 2 últimas líneas de un fichero)  
**tail -n +5** (display desde la línea 5 hasta el final)  
**| tail** (actúa también como filtro)
63. **cut** (elige las columnas a mostrar)  
**cut -d: -f3,5 empleados** (mostrar las columnas 3 y 5 con delimitador :)  
**cut -c1-3** (mostrar caracteres del 1 al 3)  
**uniq** (omitir duplicados)  
**uniq -c** (número de líneas de elementos duplicados)  
**uniq -u** (número de datos únicos)  
**uniq -i** (número de datos únicos ignorando mayus y minus)  
**cut -d: -f3,5 empleados | sort | cut -c1-3 | uniq -c** (cuantos datos diferentes existen y número de repeticiones, necesario ordenar antes)  
**cut -d: -f3,5 empleados | sort | cut -c1-3 | uniq -c | wc -l** (cuantos datos diferentes existen y número de repeticiones, necesario ordenar antes, en función de las líneas)  
**cut -d: -f3,5 empleados | sort | cut -c1-3 | uniq -u** (cuantos datos/filas únicos existen, necesario ordenar antes)
64. **paste** (pegar columnas de archivos)  
**paste -d: empleados.txt ubicacion.txt | tee empleados.local** (pegar las columnas de empleados.txt y ubicacion.txt con delimitador : y hacer display al mismo tiempo que el resultado se guarda en empleados.local)
65. **tr [A-Z] [a-z] < ubicacion.txt > ubi.minus.txt** (traductor, cambiar de mayúsculas a minúsculas lo escrito en ubicación.txt y guardarlo en ubi.minus.txt, *no es recomendable guardarlo en el mismo fichero*)  
**tr 'Ñ' '#' < ubi.minus.txt** (cambiar Ñ por # en un archivo)  
**tr -d ' ' < ubicacion.txt** (eliminar los espacios escritos en ubicación.txt)  
**tr -s ' ' < ubicacion.txt** (eliminar los espacios repetitivos en ubicación.txt)  
**tr -s [a-zA-Z] < ubicacion.txt** (traducir de minúsculas a mayúsculas en ubicación.txt)  
**| tr** (actúa también como filtro)
66. **mail** (mandar mensaje)

**mail usuario** (mandar mensaje a un usuario)

**mail fichero < usuario** (mandar aliasarchivo como mensaje a un usuario)

**?** (atributos de mail)

67. **wall** (mensaje a terminal compartida)

68. **mesg y** (activar para recibir mensajes)

**mesg n** (desactivar para recibir mensajes)

69. **last** (ver los últimos conectados)

**last -5** (ver los últimos 5 conectados)

70. **ls -l /home** (listar viendo toda la información del fichero)

71. **cd ../usuario/** (entrar en home de otro usuario)

## 72. EXPRESIONES REGULARES

72.1. **grep "texto" archivo** (busqueda de letras, palabras, conjuntos de letras, búsqueda en varios archivos archivo1 archivo2 archivo3...)

**grep -i "texto" archivo** (ignorar mayus y minus)

**grep -w "texto" archivo** (buscar palabra completa)

**grep -v "texto" archivo** (buscar lo que no contenga)

**grep -n "texto" archivo** (buscar mostrando el número de línea)

**grep -c "texto" archivo** (buscar mostrando el número de líneas en las que existe ese texto)

**grep -A "texto" archivo** (after, buscar el posterior)

**grep -B "texto" archivo** (before, buscar el anterior)

**grep -C "texto" archivo** (after and before, buscar el anterior y posterior)

**grep "..texto.." archivo** (ver 2 caracteres antes y 3 después de texto)

*/Ej. ls | grep ^...\$ (se buscan 4 caracteres)*

**grep "textx\*" archivo**

**| grep** (actúa también como filtro) */Ej. ls -a | grep "a"*

**grep "\."** (buscar los .) */Ej. ls -a | grep "\."*

**grep ^-** (buscar todo lo que empiece por -) */Ej. ls -a | grep "^-"*

**grep y\$** (buscar todo lo que termine por y)

**grep ^[A-Z]** (buscar los que empiecen por mayus)

**grep ^[^eEfF]** (buscar los que no empiecen por e, E, f, F)

**grep "\(.\\)\1"** (caracter repetido consecutivo)

**grep [[:digit:]]\$** (si termina en un dígito)

**grep [[:digit:]]** o **grep [0-9]** (buscar dígito)

**grep [[:upper:]]** (buscar mayus)

72.2. **egrep** o **grep -e**

72.3. **sed**

72.4. **awk**

72.5. **find ruta -name "texto"** (buscar en el sistema texto, NO actúa como filtro) */Ej. find ~ -name ejer8 (buscar archivo ejer8 en home)*

*/Ej. find / -name etc (buscar archivo etc en la raíz)*

**find ruta -iname "texto"** (ignora mayus y minus)  
**find ruta -iname psswd 2> /dev/null**  
**find ruta -group "grupo"** (buscar grupo o nº de grupo)  
**find ruta -user "usuario"** (buscar usuario o nº de usuario)  
**find ruta -inum "archivo"** (buscar nº de inodo)  
**find ruta -type f** (buscar por tipo de fichero ordinario)

**-f** (ordinario)  
**-d** (directorio)  
**-l** (link)  
-----  
**-b** (dispositivos bloque)  
**-c** (dispositivos de caracter)  
**-s** (socket)  
**-p** (pipe)

**find ruta -type d -perm 1755** (buscar por tipo de fichero directorio con permisos 755 y sticky bit)  
**find ruta -type d perm g=w** (buscar por tipo de fichero directorio con permiso de escritura para grupos)  
**find ruta -type f size 10k** (buscar por tipo de fichero ordinario de tamaño exacto 10k)

**10** (10k exactos)  
**+10** (>10k)  
**-10** (<10k)  
-----  
**10b** (10 bloques, cada bloque de 512)  
**10c** (10 caracter/bytes)  
**10k** (kilo)  
**10M** (10 megas)  
**10G** (10 gigas)

**find ruta -type f -atime 0** (buscar por tiempo de acceso en días, en este caso últimas 24h) /Ej. **find ruta -type f -atime 1** (buscar por tiempo de acceso hace 1 día)  
**find ruta -type f -mtime 0** (buscar por tiempo de modif en días, en este caso últimas 24h)  
**find ruta -type f -links 2** (buscar por nº de enlaces, en este caso 2)  
**find ruta -nouser** (buscar archivos sin usuario reconocido)  
**find ruta -nogroup** (buscar archivos sin grupo reconocido)  
(utilizar **2> /dev/null** para no visualizar el proceso)

**-o**    o



**-a** y

/Ej. `find .. -type f \(-size +2b -a -iname "a*" \) 2>/dev/null`

**find ruta -mindepth 2** (buscar con un mínimo de profundidad de 2)

/Ej. `find /etc -mindepth 2 \(-type f -o -type d \) -size -1M -iname "*" 2>/dev/null`

**find ruta -maxdepth 2** (buscar con un máximo de profundidad de 2)

/Ej. `find /etc -maxdepth 2 \(-type f -o -type d \) -size -1M -iname "*" 2>/dev/null`

**find ruta -mindepth 2 -maxdepth 3** (buscar con un máximo de profundidad de 2)

/Ej. `find /etc -maxdepth 2 \(-type f -o -type d \) -size -1M -iname "*" 2>/dev/null`

**find ruta -type abc123.**

**f -iname "c\*" -exec cat {} \;** (buscar con un máximo de profundidad de 2, visualizando todas las líneas con un cat y luego finalizar)

*El resultado cambia*

**find ruta -type f -iname "c\*" -exec cat {} \; | sort -n | uniq -c | tee lista.datos**  
(filtros después del ;)

**find ruta -type f -iname "\*ini" -exec head -1 {} >> ~/lista.datos.txt \;**  
(direccionamientos antes del ;)

**?** (cualquier caracter)

/Ej. `find ruta -type f -name "[Aa]dm*n" 2>/dev/nullc`

**find ruta -type f -iname "c\*" -exec cat {} \;** (buscar con un máximo de profundidad de 2, visualizando todas las líneas con un cat y luego finalizar)

**!** (negación, se puede aplicar individualmente a un parámetro)

**-not** (negación, se puede aplicar individualmente a un parámetro)

**find ruta -type f -empty** (buscar por tipo de fichero ordinario vacío, igual que un size 0)

**find ruta -type f -newer archivo** (buscar por tipo de fichero ordinario más nuevo que archivo)

73. ``` (ejecutar comando en línea) /Ej. `echo Hay `ls | wc -l` entradas en el directorio `pwd``

## 74. ALIAS

74.1. **alias** (ver los alias activos del sistema, ya sean en memoria o en el archivo .bashrc)

74.2. **alias mas="less"** (crear el alias mas con el valor igual al comando less, se guardará en memoria; también se puede sobrescribir uno ya creado)

74.3. **unalias mas** (eliminar el alias mas creado en memoria)

**~/.bashrc** (archivo donde se pueden guardar alias en el apartado  
**# some more ls alisases**, se encuentra en ~ como archivo oculto)

## 75. VARIABLES DE ENTORNO



- 75.1. **env** (visualizar las variables de entorno del sistema que han sido exportadas)
- 75.2. **set** (visualizar todas las variables de entorno del sistema)
- 75.3. **unset** (eliminar de las variables de entorno del sistema)  
*para visualizar variables de entorno concretas (mirar [echo](#))*
- 75.4. **nombre="Xesús López"** (visualizar las variables de entorno del sistema)

**\$BASH\_VERSION**

**\$USER**

- - - - -